Forecasts of Realized Volatility

Xiangtian Deng*

December 14, 2019

Abstract

In this project, we mainly use different methods to forecast the future realized volatility of several indexes. We compare performances of Har model, Rough Volatility model and neural network. All these method have strong predictive power and performances of them are highly dependent on the index we choose.

1 Introduction

1.1 Har Model

In Har model[1], we try to model the long memory of realized volatility. The idea is to use later realized volatilities to regress on previous realized volatilities. We denote daily realized volatility with RV_t^d. Also, in our notation, a weekly realized volatility at time t is given by the average

$$RV_t^{(w)} = \frac{1}{5}(RV_t^d + RV_{t-1d}^d + ... + RV_{t-4d}^d). \tag{1.1}$$

Similarly we can define monthly realized volatility $RV_t^{(m)}$ at time t as

$$RV_t^{(m)} = \frac{1}{22}(RV_t^d + RV_{t-1d}^d + ... + RV_{t-21d}^d) \tag{1.2}$$

and even arbitrary time period volatility.

Then we model future realized volatility using following formula,

$$RV_{t+\Delta}^{(d)} = c_{(\Delta)} + \phi_\Delta^{(d)} RV_t^{(d)} + \phi_\Delta^{(w)} RV_t^{(w)} + \phi_\Delta^{(m)} RV_t^{(m)} + \epsilon_{t+\Delta}, \tag{1.3}$$

where ϵ is the noise term of of the regression. This formula, however, may give us a negative forecast of realized volatility, so we can do the following modification.

$$\log RV_{t+\Delta}^{(d)} = c'_{(\Delta)} + \psi_\Delta^{(d)} \log RV_t^{(d)} + \psi_\Delta^{(w)} \log RV_t^{(w)} + \psi_\Delta^{(m)} \log RV_t^{(m)} + \epsilon'_{t+\Delta} \tag{1.4}$$

*Department of Mathematics, Baruch College, CUNY. forest.deng1995@gmail.com

1.2 Rough Volatility

In rough volatility model[2], we use fractional brownian motion to model volatility.

$$\sigma_t = \sigma e^{\{\nu W_t^H\}}. \tag{1.5}$$

Here W_t^H is a fractional brownian motion with Hurst parameter H.

$$\log \sigma_{t+\Delta} - \log \sigma_t = \nu(W_{t+\Delta}^H - W_t^H). \tag{1.6}$$

According to the property of fractional brownian motion, volatility has following property

$$\mathbb{E}[|\log \sigma_{t+\Delta} - \log \sigma_t|^q] = C^q \Delta^{qH} \tag{1.7}$$

.

To determine H, we can use different Δ to calculate $m(q, \Delta) = \mathbb{E}[|\log \sigma_{t+\Delta} - \log \sigma_t|^q]$ with different q. Then, for a certain q, we can fit power law curve to $(\Delta, m(q, \Delta))$ get the power $p(q)$ and do linear regression to $(q, p(q))$ to get Hurst parameter H.

After we get the value of H, we can then use the following formula to forecast future volatility.

$$\mathbb{E}[\log \sigma_{t+\Delta}|\mathcal{F}_t] \approx \frac{1}{A}\{\frac{\log \sigma_t}{(s^\star + \Delta)(s^\star)^{H+\frac{1}{2}}} + \sum_{j=1}^{\infty} \frac{\log \sigma_{t-j}}{(j+\frac{1}{2}+\Delta)(j+\frac{1}{2})^{H+\frac{1}{2}}}\}, \tag{1.8}$$

where

$$A = \frac{1}{s^{\star H+\frac{3}{2}}} + \sum_{j=1}^{\infty} \frac{1}{(j+\frac{1}{2}+\Delta)(j+\frac{1}{2})^{H+\frac{1}{2}}}, \tag{1.9}$$

and

$$s^\star = \gamma^{\frac{1}{1-\gamma}}. \tag{1.10}$$

Here $\gamma = \frac{1}{2} - H$.

1.3 AR Model

Recall in the forward forecast formula (1.8), the conditional forward $\log \sigma$ is a linear combination of historical $\log \sigma$. A natural idea is that instead of directly using rough volatility model, we can use AR model to see if we can get the same kernel as that in the rough volatility model.

AR model we used is as follows,

$$\log \sigma_{t+1} = c + \sum_{i=0}^{M} \psi^i \log \sigma_{t-i} + \epsilon_t, \tag{1.11}$$

where ψ_i is the AR kernel and ϵ_t is the noise term in AR model.

1.4 Neural Network

The methods mentioned before are all linear method. Here, we use a non-linear method to do forecast, which is Neural Network. Normally the output of neural net for regression can give only one output. Here we modify the traditional neural network a little bit to make it able to give multiple outputs (as in Table 1).

Input (latest 200 realized volatility)
Hidden Layer
Output (100 forward realized volatilities from $t+1$ to $t+100$)

Table 1: Structure of Neural Net

For the reason that outputs of the neural network is dependent, we try different versions of loss functions to train the neural network. Let δ_i denote the difference between the true values of 100 outputs and the forecasted values of 100 outputs at time t. In other words,

$$\delta_i = \begin{bmatrix} \log\sigma_{t+1} \\ \log\sigma_{t+2} \\ \dots \\ \log\sigma_{t+100} \end{bmatrix} - \begin{bmatrix} \widehat{\log\sigma_{t+1}} \\ \widehat{\log\sigma_{t+2}} \\ \dots \\ \widehat{\log\sigma_{t+100}} \end{bmatrix}, \tag{1.12}$$

where $\log\hat{\sigma}_{t+i}$ is the value of log volatility at time $t+i$ forecasted by the neural network, and we use Σ to denote the covariance matrix of multiple forward log realized volatilities. In other words,

$$\Sigma = \text{cov}\left(\begin{bmatrix} \log\sigma_{t+1} \\ \log\sigma_{t+2} \\ \dots \\ \log\sigma_{t+100} \end{bmatrix}\right)) \tag{1.13}$$

The loss functions we choose are

1.
$$f_1(\theta; \log\sigma) = \overline{\delta^T\delta} = \frac{1}{N}\sum_{i=1}^{N}\delta_i^T\delta_i \tag{1.14}$$

2.
$$f_2(\theta; \log\sigma) = \overline{\delta^T\Sigma^{-1}\delta} = \frac{1}{N}\sum_{i=1}^{N}\delta_i^T\Sigma^{-1}\delta_i \tag{1.15}$$

2 Numerical Results

In this section, we use data from [3] to compare different forecast methods mentioned in the previous part.

2.1 Comparison between Har model and Rough Volatility model

In this part, we use 28 different indexes to compare accurracies of forecasts made by Har model and rough volatility models. We use data from Jan. 3rd, 2000 to Oct. 11th, 2011 to get the Hurst parameter, H, in rough volatility and the regression coefficients in Har model. Then, we use data from Oct. 12th, 2011 to Oct, 12 2019 to test these two models.

We also choose a benchmark, unconditional forecast, which regards volatility process as a markov process and the forecast is

$$\widehat{\log \sigma_{t+\Delta}}^{unconditional} = \log \sigma_t. \tag{2.1}$$

The result of the comparison is shown in Figure 1, where each subplot is a comparison between Har model and rough volatility model in a certain index. The x-axis is the time lag (Δ in 2.1). The y-axis is the mean square error of forecasts.

From Figure 1, the performance of Har and rough volatility model are all better than the naive unconditional forecast, and there are no major difference between Har model and rough vol model in terms of perfermance.

2.2 Kernel Comparison between Rough Volatility Model and AR Model

Recall that the forecast of future log realized volatility is just linear combination of historical log volatility. In this part, we fit AR model and to see if the AR kernel is roughly the same as rough vol kernel ($\Delta = 1$ in 1.8). Comparison is implemented only on SPX realized volatility data.

The result is shown in Figure 2. The AR kernel is fluctuating around rough volatility kernel. In other words, rough volatility kernel can be regarded as a kernel smooth version of AR kernel, which indicates that rough volatility model is very close to the true dynamics of volatility.

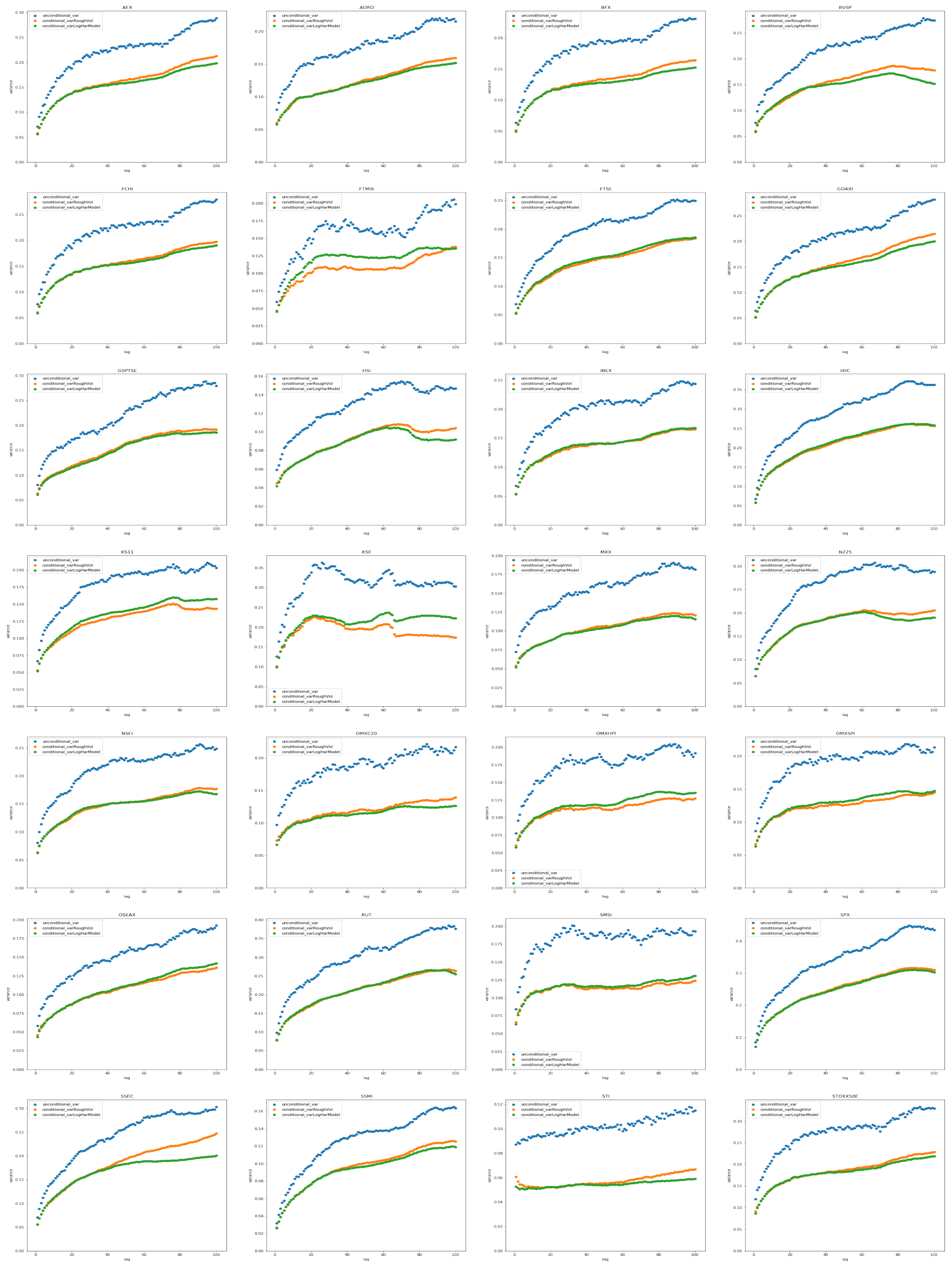
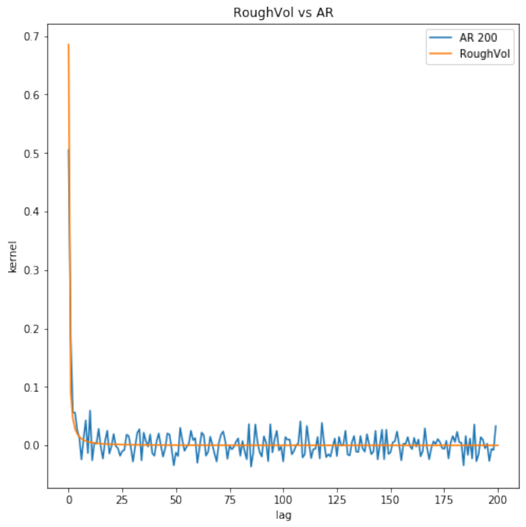


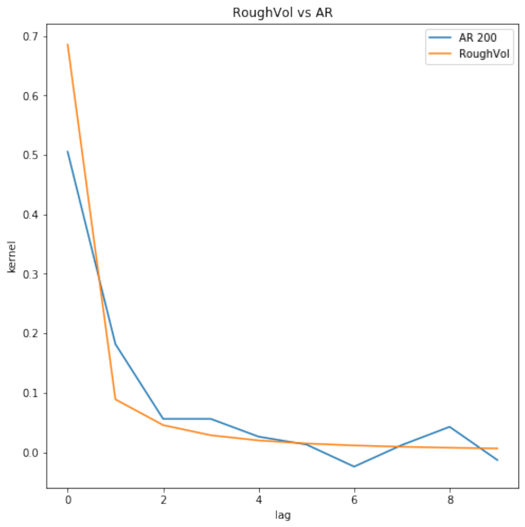
Figure 1: Comparison between Har model and Rough Volatility Model

2.3 Neural Network Forecast

In this part, we use neural network to forecast future log volatility and hope this can give a better result than rough volatility model. We use more data to train models,



(a) 200 lags



(b) first 10 lags

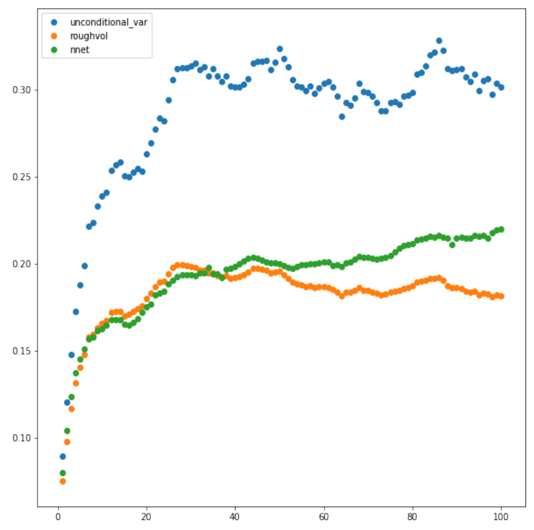
Figure 2: Kernel Comparison between Har Model and Rough Volatility Model

because Neural Network can be easily overfitted. We use data from Jan. 3rd, 2000 to Oct. 11th, 2015 to train the neural network and get the Hurst parameter, H, in rough volatility. Then, we use data from Oct. 12th, 2015 to Oct, 12 2019 to test these two models.

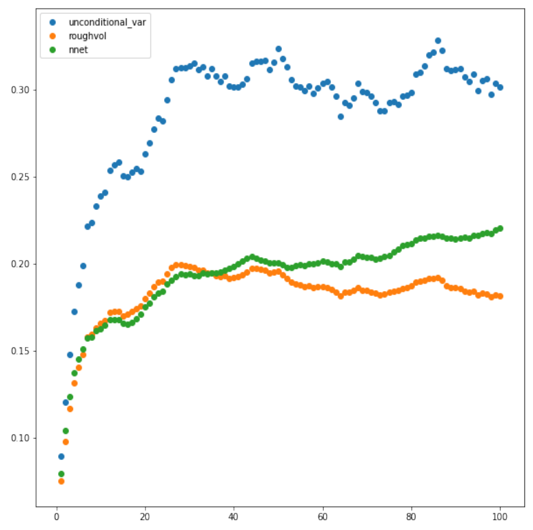
Firstly, We compare both loss functions in 1.14 and 1.15 with SPX data as shown in Figure 3. The test idea is the same as the idea used in Figure 1.

As we can see in the figure, neural network trained by these two loss functions are slightly worse than rough volatility model and two loss functions give similar results. As a result, in the following analysis, we use loss function 1.14 to train our neural network to make everything simpler.

Next, to see the performance of neural network and rough volatility model in different situations, we use four other indexes to compare neural network and rough volatility model. The result is shown in Figure 4. In terms of these four indexes, neural network performs better than rough volatility. If we have a closer look at BVSP, the performance of neural network is much better than rough volatility. The reason is that if we have a look at unconditional error of BVSP. The unconditional error is just $\overline{(\log \sigma_t + \Delta - \log \sigma_t)^2}$ which should be power law as Δ increases. Unconditional error of BVSP however is clearly far away from power law, which means the realized volatility mechanics of BVSP is not rough volatility. In this kind of situation, performance of neural network will be better than rough volatility model.



(a) Loss 1



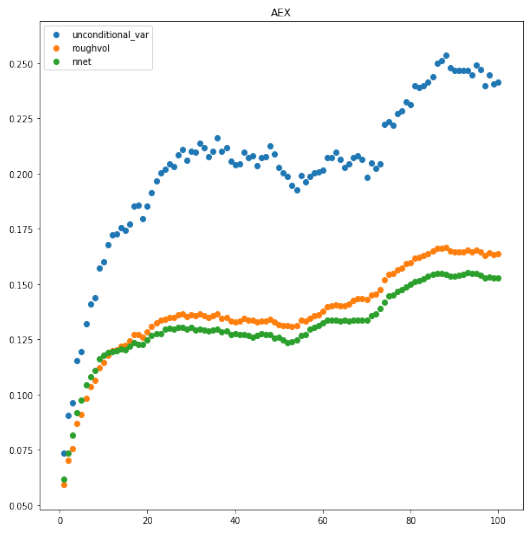
(b) Loss 2

Figure 3: Loss Function Comparison with SPX Realized Volatility

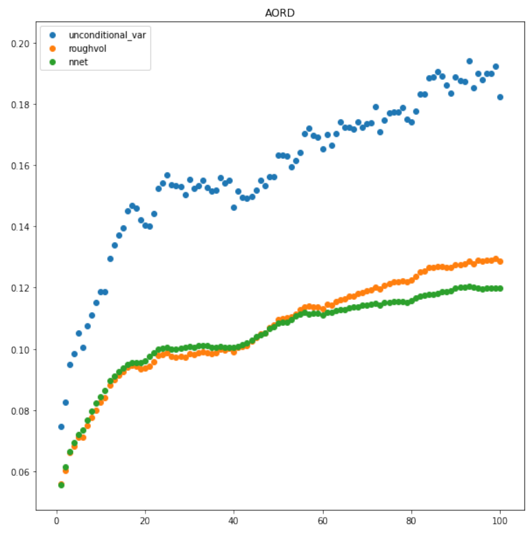
3 Summary and Conclusion

Using daily realized volatility data, we realized that Har model, rough volatility model and neural network all had very strong predictive power compared to the naive unconditional predictor (2.1). Although performances of these three methods are highly dependent on the index we choose, these three methods perform similarly in most situations.

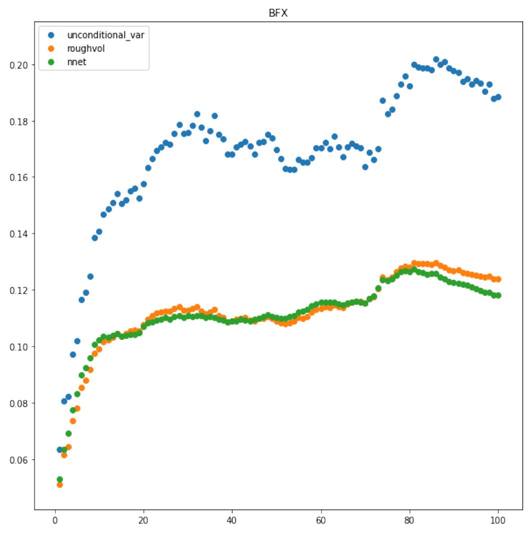
The kernel of rough volatility model is very similar to the kernel of AR model, which indicates that the underlying mechanics of realized volatility is very close to rough volatility model. For some indexes, BVSP for example, the dynamics of realized volatility, however, is too far away from rough volatility. As a result, the performance of rough volatility models will significantly worse than other models in these kind of indexes.



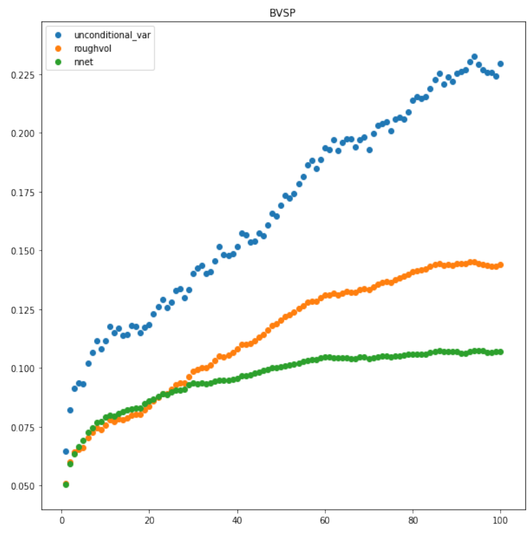
(a) AEX



(b) AORD



(c) BFX



(d) BVSP

Figure 4: Neural Network vs Rough Volatility

Acknowledgments

I am very grateful to Prof. Jim Gatheral who give me very strong support during my capstone project. I also need to say thank you to Kathy Jin who help me proofread my report and give me some excellent ideas.

References

[1] Corsi, F., 2009. A simple approximate long-memory model of realized volatility. Journal of Financial Econometrics, 7(2), pp.174-196.

[2] Gatheral, J., Jaisson, T. and Rosenbaum, M., 2018. Volatility is rough. Quantitative Finance, 18(6), pp.933-949.

[3] Oxford-Man Institute of Quantitative Finance, Realized Library, https://realized.oxford-man.ox.ac.uk/about-us

[4] Gatheral, J., 2011. The volatility surface: a practitioner's guide (Vol. 357). John Wiley & Sons.

[5] Goodfellow, I., Bengio, Y. and Courville, A., 2016. Deep learning. MIT press.

[6] Ketkar, N., 2017. Introduction to pytorch. In Deep learning with python (pp. 195-208). Apress, Berkeley, CA.